An Efficient Storage Mechanism for Non Reference Elements in Cloud Utilizing Tversky Co-Efficient, K-Means and SVM

Shweta Pandey , Sunil Shukla

Abstract*: *Topical technological advances have prompted the success and popularity of the cloud. This novel hypothesis is becoming more interesting as it provides economical architectures that support storage, transmission,and intensive data computing. Though, these capable storage services pose many challenging design problems because of data control loss with cloud's abstract nature. The aspiration behind this research is to propose a novel mechanism that could lessen the cloud storage using Tversky Co-efficient technique by means of K-mean clustering. For the cross-validation process, the SVM classification algorithm is considered. For the computation, QoS parameters, such as Formation Time and Formation Time are considered that would enlighten the efficacy of the research work.

Index Terms: Cloud computing, cloud storage, Tversky Co-efficient, SVM, Formation Time, Formation Time

I. INTRODUCTION

Cloud computing is a dispersed computing model that behaves as an intermediate later among cloud data centers and IoT (internet of things) sensors or devices [1]. It provides networking, computing and storage services so that the cloud-based services could be enhanced quickly for IoT sensor or the devices. Cloud computing was inked by Cisco in 2012 for addressing IoT challenges applications in traditional cloud computing [2].

The environment of Cloud computing consists of state-of-art components of networking, viz. switches, routers, proxy servers, set up boxes and BS (Base Stations) that could be placed at the adjacent proximity of sensors and devices as depicted in the above figure. As shown, the components are given having storage, distinct computing and networking abilities that could support the execution of service applications. Though, the networking components facilitate Cloud computing for creating huge geographical distributions of the cloud services. Cloud computing may execute effectively by means of power consumption, service latency, network traffic,and content sharing and other related services [3].

Fig.1depicts the Cloud system with three-tier architecture, viz. cloud, IoT or end users and Cloud stratums. The Cloud stratum has been developed by a number of cloud domains authorized by similar or diverse providers. The domains are produced by Cloud nodes with switches, edge routes, PCs, access points. IoT or end users are produced by two domains, initial one is end-user device and subsequent is IoT device. It has to notice that in the stratum, one of the two domains might be absent. The communication among IoT or end users and the Cloud stratum is done via LAN (Local area network). Though, the communication needs a connection in WAN (Wireless area network) via Cloud [4].

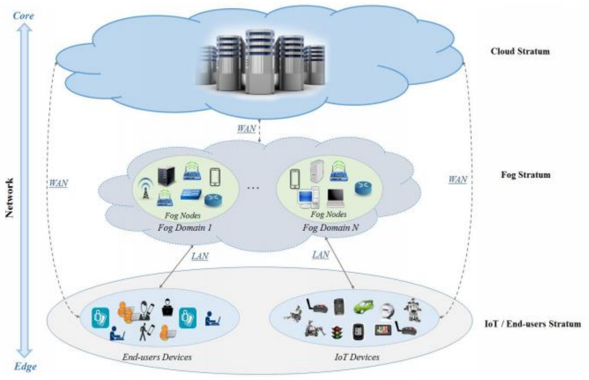

Fig.1: Cloud computing environment

Cloud computing is known as a promising expansion of the cloud computing model for handling IoT issues at the core of the network [5]. Though, in Cloud computing, the processing nodes are distributed and heterogeneous. In addition, the services of Cloud computing have to treat the varied constrained environment features. The assertion of security is also effective in Cloud computing. Though, it faces a number of challenges as demonstrated beneath [6]:

a) Dynamic and Fluctuating workloads

The main challenge for fluctuation in fog computing is that the workloads are random. The fluctuation of workload can occur in a pre-planned or post-planned manner. In pre-planned fluctuation, the circumstances could be predicted before, so that the resource may be allocated efficiently or not and on exact time span.

b) Ensuring efficient resource utilization

The resources have to beplaced directly whenever required while it is post planned and it is known as auto-allocation in fog computing. The arriving workload must bedesignated to the resources such that the scheduler of the fog computing has to guarantee effective resource utilization. It needs improved scheduling techniques for the task allowance for accessible processors.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Shweta*, Computer Science Department, Deshbhagat University.
Dr. Sunil Shukla, HOD Computer Science Department, Deshbhagat University.

.

Table 1. Cloud Computing Challenges

Challenges	Narration
Structural issues	Varied components from the core and edge network could be utilized as an effective Cloud computing architecture.
	Generally, these components are occupied with different processors but are not considered for typical computing. Providing the components with common computation than their existing activities would be a challenging task.
	The suitable node selection on the basis of executive and operational requirements, equivalent resource configuration,anddeployment sections are indispensable in Cloud.
	The computational nodes are dispersed at the edge network and could be virtualized.
	Therefore, detection of suitable metrics, methods for intermodal collaboration and effective resources are significant.
Service-oriented	The policies to disperse computational tasks and the services between IoT sensors/devices, cloud and Cloud infrastructure are needed to be defined. Data visualization via web interfaces is also tough for designing.
	SLA (Service level agreements) is usually affected by different factors like energy, service cost, characteristics of applications, network status,and data flow.
	So, on exact scenario, it is tough to define the service provisioning metrics and equivalent SLOs (Service level objectives).
Security aspects	As Cloud computing is designed on existing networking components, so, it is more susceptible to the security attacks.
	The authenticated access for the services and privacy maintenance in a huge dispersed model like fig computing are difficult to ensure.
	The execution of the security mechanism for data-centric integrity may influence the QoS of Cloud computing to large extent.

A. Storage Issues of Cloud

Cloud is a paid structure and hence what is getting stored at the network plays a vital role in whatuser going to pay [7]. This paper focuses on the development of a novel structure which not only protects the data but also keeps the storage element in mind. The storage section of Cloud is also termed as Data Base As a Service (DAAS) [8].

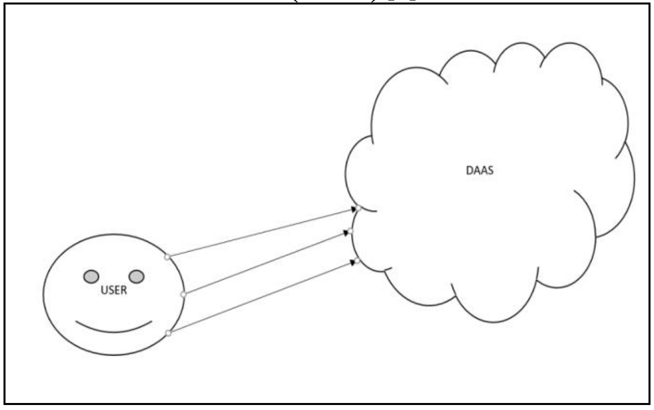

Fig.2 User to Cloud

The user sends it data to a cloud server and it becomes the responsibility of the cloud server to store the data which not only reduces the storage cost but also provides the reliability [9]. The proposed paper has worked on two areas namely Data Security and non clustered data storage. The organization of the manuscript is such that section II represents the proposed work, section III evaluates the proposed work and Section IV draws the crux of the paper.

II. PROPOSED WORK MODEL

The proposed work is segregated in two sections namely relation evaluation with and utilization of Cross-reference checker utilizing Support Vector Machine(SVM) [10].As mentioned earlier the proposed work architecture considers the non clustered environment that considers no previous clusters available. Hence the clustering will be done only as per the available data elements. To do so, Tversky Co-relation is evaluated between the data elements sent to the cloud [11].

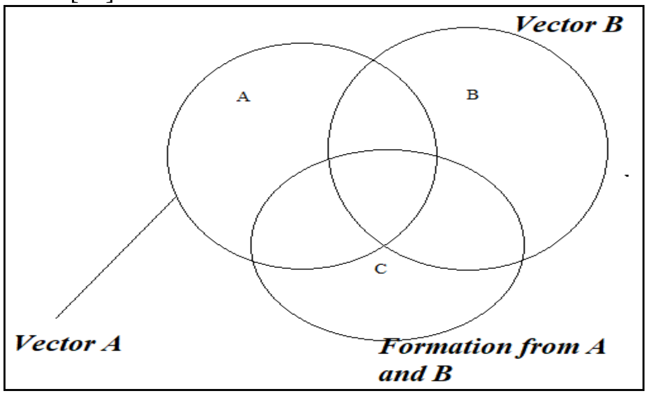

Fig.3 Evaluation Basic of Tversky Co-efficient

Writing in equation form, this coefficient can be written as follows:

$$T(A,B) = \frac{|A\ Common\ \ B|}{|-A+B| + x|A-B| + n|A*B|} \qquad (1)$$

Where A and B are documented vectors and x and n are constants which are combined with A and B to form C.Algorithm1 evaluates the Tversky coefficient (TC) utilizing (1) [12].

Algorithm 1: Calculation of Tversky Co-efficient

```
1.  Tversky = functionTverskySimilarity(docs)
      The input of this calculator is the numeral vector of
      each data file
2.  //Tversky_sim = []; // Empty array for the
      calculation of similarity index
3.  Sim_count = 0; // Total number of identified
      similarities
4.  For m = 0 todocs.length // For 1 to total number of
      data files
5.  Current_doc = docs (m); // Doc I
6.  For n = I + 1 todocs.length // Next data value
      series
7.  L=|Tversky(current_doc)-Tversky(docs(n))|;
8.  Tversky_sim[sim_count, 0] = current_doc; //
      The similarity measure has three columns
9.  Tversky_sim[sim_count, 1] = docs(n); // First
      column is the main data file , Second Column is
10.  The//Connecting data file
11.  Tversky_sim[sim_count, 2] = L; // The
      similarity value
12.  Sim_count = Sim_count + 1; // Counter
      incremented by 1
13.  Endfor
14.  Endfor
15.  End function;
```

The TC forms n number of clusters utilizing K means clustering algorithm. The objective is the partitioning of objects in the form of clusters that should consider the observations as the clusters including the nearest mean. The clusters embrace the Centroid or the cluster head which is equivalent to the cluster point average. The clustersare chosen arbitrarily by the user [13]. It carries with the allocating points with the repetitive manner in the closest cluster which is computed by the Euclidean distance.Measuring distance between text documents, given two documents d_a and $d_b$denoted by their term vectors $\vec{t_a}$and $\vec{t_b}$correspondingly, the Euclidean distance concerning the two documents is evaluated as:

$D_E(\vec{t_a},\vec{t_b}) = (\sum_{t=1}^{m}|w_{t,a} - w_{t,b}|^2)^{1/2}$ (2)

Here D is calculated utilizing TC mentioned in Algorithm 1. sWhere, the term set is $T = \{t1,...,tm\}$.As mentioned previously, $tfidf$value is considered as term weights, that is $w_{t,a} = tfidf(d_a, t)$.

a) Select the k cluster centers for coinciding by means of k arbitrarily selected patterns or selecting the points within the hypervolume with the pattern set.
b) Allocate every pattern to the nearby cluster center.
c) The computation of the cluster center with the existing cluster memberships is executed again.
d) Consider step two if the convergence principle is not achieved.

K-means Convergence criterion function

$E = \sum_{i=1}^{k}\sum_{j=1}^{n_i}\|x_{ij} - m_i\|^2$ (3)

Wherethe ith cluster has x_{ij} with jthdata point;m_i is theiith cluster centerandn_i is the number of data points of ith cluster.

The formed clusters are cross-validated utilizing SVM [14]. SVM takes the TC of each placed cluster element as cross-validation input and performs a supervised learning architecture mechanism to verify the amendment of the clusters. A polynomial kernel is used for Kernel verification [15].

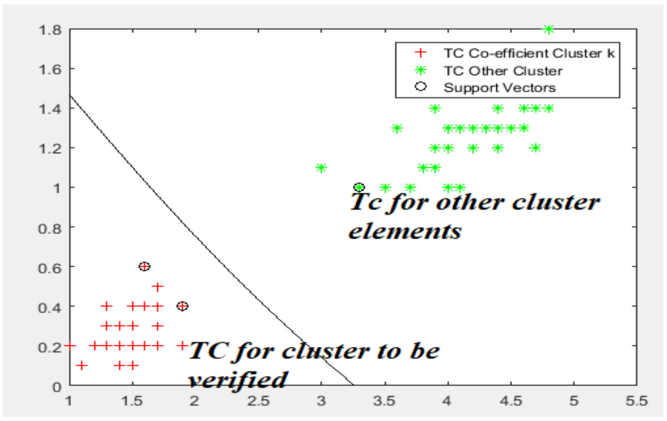

Fig.4 SVM Training Architecture

For each group formed utilizing K means in evaluated using SVM with respect to the relative group value containing TC value. The supervised mechanism takes the output of training as the test input and cross-validates the alteration of the clusters [16].

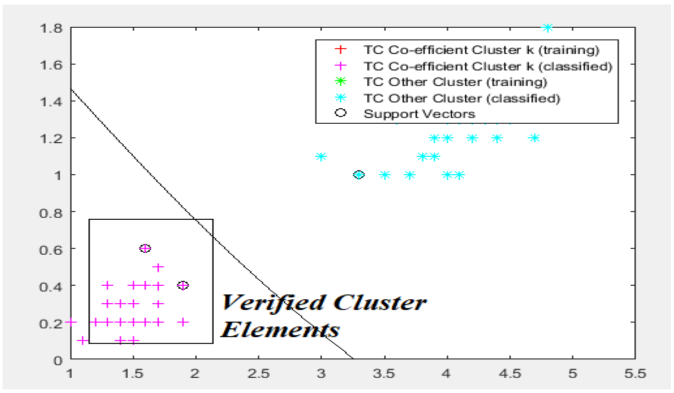

Fig.5 Cross Validated Structure

Based on the formed structure, the service parameters are evaluated which is discussed in Section III. The development of the proposed algorithm is done on NETBEANS with the polynomial kernel for SVM.

III. RESULTS EVALUATION

The results are evaluated based on the formation time and Cloud Cost consumption based on Time Consumed information.

a) Formation Time (FT): It is the total time consumed on the formation of the Cloud Cluster. It varies with the size of supplied files.

$$FT = Consumed\ Time\ /\ Data\ Size \quad (4)$$

b) Cost of Formation (CF): It is the total cost of Cloud Cluster formation. It depends upon two things namely the total supplied data files and FT.

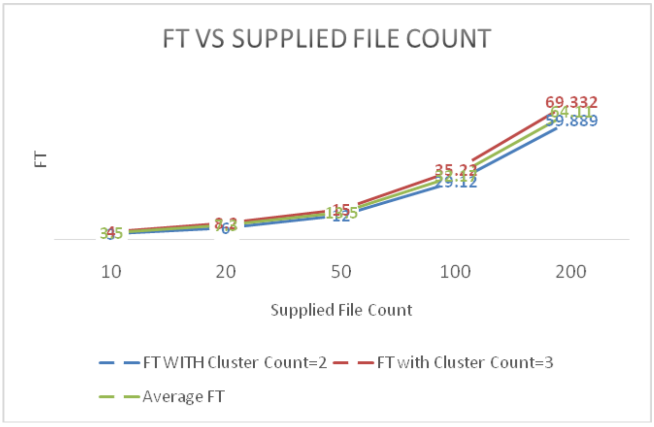

Fig.6 CF vs Supplied File Count

Fig.6 illustrates the performance of the proposed principle based on the FT. It is obvious that when the file count will increase, the total formation time will increase. The proposed work has taken two considerations namely FT with 2 clusters and FT with 3 clusters. An average FT is also demonstrated. For a total supplied file count of 200, 2 cluster mechanism consumed 59.889 seconds whereas 3 cluster mechanism consumes 69.332 seconds. The average consumption time will be $\frac{59.889+69.332}{2} = 64.11$ seconds. The CF is directly proportional to FT. Hence CF is maximum for 200 data files.

IV. CONCLUSION

This paper aims to reduce the cloud storage costing by implementing a new cluster formation technique based on the Tversky co-efficient similarity index followed by K means clustering. The paper considers a non reference attribute based clustering in which is no previous reference is present for any document file and hence the similarity based storage management is presented. The formed clusters are also cross-validated utilizing SVM which takes the linear polynomial and supervised learning mechanism. The proposed mechanism is evaluated using Formation Time and Formation Cost. Almost 200 files were supplied at max considering that the developed cloud is a small scale cloud for now. The maximum consumed time for 200 files with a 3 cluster mechanism is 64.11 seconds whereas for a 2 cluster mechanism it is 59.889 seconds. The currently employed work leaves a lot of future possibilities. The cloud, in this case, is considered as small scale cloud and hence only a total of 200 files were tested. The same architecture can be tested for large scale cloud which contains more than 1000 files. This is also lead to the formation of Big Data in Cloud Computing. In addition to this Neural Network can also be applied instead of SVM. Secure visions can also be added in the proposed architecture.

REFERENCES

1. Kaleeswari, C., Maheswari, P., Kuppusamy, K., &Jeyabalu, M. (2018). A Brief Review on Cloud Security Scenarios.
2. Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. Future Generation Computer Systems, 78, 964-975.
3. Fu, J., Liu, Y., Chao, H. C., Bhargava, B., & Zhang, Z. (2018). Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing. IEEE Transactions on Industrial Informatics.
4. Atan, R., Talib, A. M., & Murad, M. A. A. (2018). Formulating a security layer of cloud data storage framework based on multi-agent system architecture. GSTF Journal on Computing (JoC), 1(1).
5. Jose, G. S. S., & Christopher, C. S. (2018). Secure cloud data storage approach in e-learning systems. Cluster Computing, 1-6.
6. Alsmadi, D., &Prybutok, V. (2018). Sharing and storage behavior via cloud computing: Security and privacy in research and practice. Computers in Human Behavior, 85, 218-226.
7. Akherfi, K., Gerndt, M., &Harroud, H. (2018). Mobile cloud computing for computation offloading: Issues and challenges. Applied computing and informatics, 14(1), 1-16.
8. Aliyu, A., Abdullah, A. H., Kaiwartya, O., Cao, Y., Usman, M. J., Kumar, S., ... & Raw, R. S. (2018). Cloud computing in VANETs: architecture, taxonomy, and challenges. IETE Technical Review, 35(5), 523-547.
9. Geeta, C. M., Raghavendra, S., Buyya, R., Venugopal, K. R., Iyengar, S. S., & Patnaik, L. M. (2018). Data Auditing and Security in Cloud Computing: Issues, Challenges and Future Directions. International
10. Radha, S., &Babu, C. (2018). An Enhancement of Cloud Based Sentiment Analysis and BDAAs Using SVM Based Lexicon Dictionary and Adaptive Resource Scheduling. Journal of Computational and Theoretical Nanoscience, 15(2), 437-445.
11. Jimenez, S., Becerra, C., &Gelbukh, A. (2013). Softcardinality-core: Improving text overlap with distributional measures for semantic textual similarity. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity (Vol. 1, pp. 194-201).